REMOTE BATCH ENTRY SYSTEM UTILIZING MICROCOMPUTER AND TELEVISION DISPLAY

Richard John Simpson

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

REMOTE BATCH ENTRY SYSTEM
UTILIZING MICROCOMPUTER
AND
TELEVISION DISPLAY

by

Richard John Simpson

Thesis Advisor: R. H. Brubaker

March 1974

Remote Batch Entry System
Utilizing Microcomputer
and
Television Display

by

Richard John Simpson
Lieutenant Commander, United States Navy
B. S. Naval Postgraduate School, 1973

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the
NAVAL POSTGRADUATE SCHOOL
March 1974

ABSTRACT

A project to develop a Ring telecommunication network is currently underway at the Naval Postgraduate School. A Remote Batch Entry system is being developed as a part of that project. This paper investigates the protocol for such a system and proposes a smart CRT terminal for use as a remote station. It first develops the requirements for such a terminal and then presents the actual implementation of a TTY-like CRT terminal as the foundation in the development of a smart terminal.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

A . BACKGROUND

During the past year the personnel at the Naval Postgraduate School have been investigating the proposal to establish a local ring-structure communication network. It will provide a link between present and future computers at NPS. In conjunction with this endeavor is the development of a remote entry system to operate on the ring, permitting access to the various computer systems. This paper proposes a smart front-end visual display terminal which will utilize microcomputer technology for flexibility. The terminal system, here after referred to as the RBES (Remote Batch Entry System), is an attempt to provide the user with an effective, easy to use system. The object of this development is not to duplicate a time sharing system, but rather to extract as many of the advantages of such a system as possible, while eliminating all or most of the costly and unnecessary characteristics. The RBES proposal is an attempt to bridge the gap between batch and timesharing.

B . PURPOSE

The primary goal of this system is to provide a low cost, reliable, remote terminal which can offer some of the advantages of a timesharing terminal without requiring the constant support of a large processor. The terminal should be compatible with the ring network. Since it utilizes an Intellec-8 for its local processing, the actual coupling to the ring interface becomes a simple programming problem.

The paper will be presented in two phases. The first will outline the requirements for the development of a

"Smart Terminal"(i.e. one which can perform some of the functions of the main-frame computer, independent of it). Phase two presents the implementation of the components of the RBES as a full duplex TTY-like CRT terminal. This approach is necessary to demonstrate the reliability and usefulness of the components and to provide a basis from which the actual "Smart Terminal" might evolve.

The microcomputer program for phase 2 of the RBES can be written directly into object code or, more easily into a higher-level language. For overall convenience, PLM, a higher-level language specifically designed for Intel's MCS-8, was chosen. PLM is operational on the IBM-360. It is designed to afford the programmer the convenience of the high level language yet it retains most of the control and efficiency of an assembly level language. The program, stored in ROM (read only memory), consists of a series of subroutines, one for each different terminal command defined by the user, and a master routine to control the branching to the appropriate routine.

To keep cost down and maintain a high degree of simplicity, off-the-shelf components were utilized to the maximum degree possible. A TV interface was constructed using the Intel AT-185 design. Modifications were made to make it MCS-8 compatible. The capabilities of the terminal were programmed into MCS-8 code utilizing PLM, the higher level language described earlier. Microcomputers were chosen for a variety of reasons; but the two most prevelent are, low cost and flexibility.

II. GENERAL DESCRIPTION

A. CHARACTERISTICS OF RING NETWORK

1. Ring Structure

The ring system is of the type shown in figure 1. It is uni-directional with one node (consisting of a host processor and its Ring Interface) in control of the ring at a time. An active node is defined as a host processor with a process (at least one) capable of using the ring as a communication's medium. A processor may have more than one process capable of ring communication. The respective RI is "loaded" with the names of the processes active at a particular host during the start-up phase. Loading of process names is accomplished by providing an address to the Process Name Memory (PNM) in the Ring Interface. The bit determined by this address is set to a '1' to indicate an active process name. Active process names are deleted by resetting this bit.

A node gains control of the ring when a control token (CTL) is received (tokens are 4-bit characters in violation of the coding scheme and are used for control and synchronization). The node in control will propagate the control token after optionally transmitting a message. Nodes not in control are monitoring the ring for a CTL or SOM (start-of-message) token.

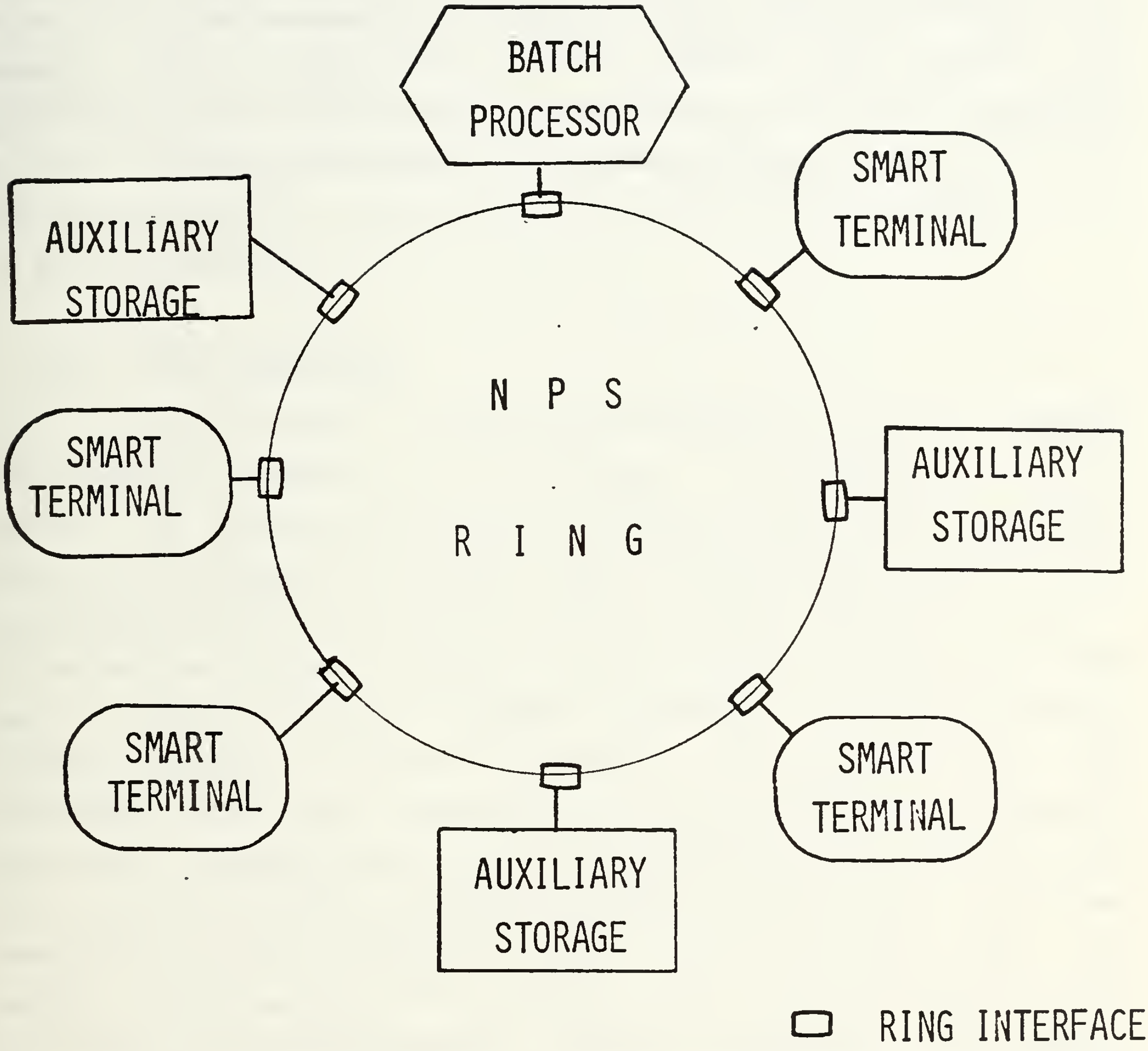


Figure 1. NPS Ring

Each node must examine each message header to determine if the destination code matches a previously loaded host process name. Each node forwards the message to the next node even if a destination match occurs. Messages are removed from the ring by the source node. This insures messages complete a circuit of the ring once and only once. Receipt of the message is signified by use of match and accept bits appended to the message and altered by destination ring interface. These bits are made available to the sending process and a determination is then made by the originator (in the host) as to whether the message is to be re-transmitted.

2. <u>Ring Interface</u>

The purpose of the Ring Interface is to 1) reshape and forward signals on the ring, 2) recognize control tokens, 3) compare destination process names with processes active on its host, 4) signal host and assemble bits of the incoming message into 8-bit bytes and pass them to the host, 5) determine whether overrun has occured, 6) check CRC bits for an error in the received message, 7) set the match, accept, and bad CRC bits accordingly, 8) replace the CTL violation with an SOM when its host process has a message ready for transmission, 9) switch outgoing messages onto the ring, 10) calculate and insert CRC bits onto the ring after generating an EOM violation after text transmission, 11) output a '0' match, a '0' accept, and a '0' bad CRC bit for resetting by the destination process' interface, 12) examine returning message status bits and make them available to the host for examination, and 13) maintain the ring active with SYN (synchronization) or CTL violations when in a master status. In addition the RI plays a significant role in ring synchronization and error recovery. Communication with the Ring Interface is accomplished using two input and two output ports. One input and one output port are used for

message transfer with the two remaining ports used for control and status information.

B. SUPPORT SYSTEMS

An accurate understanding of this particular design can only be achieved if certain future components can be considered as "black boxes". The following assumptions are by no means extraordinary, but rather are basic characteristics of systems slated for the future.

1. Auxiliary Storage (AUX)

The auxiliary storage system, here after referred to as AUX, is a collection of one or more storage facilities each connected to the ring by its own RI and have the following capabilities.

a. Ability to interpret messages received from any process requiring file access (e.g. Smart Terminal, Batch Processor).

b. Capability of maintaining a catolog of file names, user ID's, passwords and pertinent file characteristics.

c. Facility to store received data in such a manner that line by line fetch and replacement is realized.

d. Power to provide between the line insertion of additional data or deletion of existing lines (e.g. a special 'Delete Line' mark could be utilized rather than file compression).

e. Creation and deletion of files.

f. Capability to acquire initial and/or additional storage space when requested, or facility to compose and transmit warning message when not available.

2. Batch Processor (BP)

The batch processor can be one or a collection of

processors each having its own interface to the ring network. However, for simplicity's sake, the BP reference is to just one batch processor. The BP will have the following basic capabilities:

a. It will interpret and compose messages from and to AUX and Smart Terminal.

b. It should be able to prohibit unqualified intrusion into files by requiring user ID and password.

c. It shall obtain programs and data from AUX and execute when requested by a process on the ring (e.g. a Smart Terminal).

d. It should have the ability to transfer the output of a program to AUX and notify the requesting process (e.g. Smart Terminal).

C. SMART TERMINAL (ST)

The expression "Smart Terminal" can be simply defined as a terminal which can perform various degrees of local processing. An example of this is file editing, which reduces the time the user must be connected to the processing computer. A smart terminal does less than a total processing job, but it can take over jobs formerly done only by the mainframe saving valuable mainframe time. The proposed smart terminal for the NPS Ring network will initially have the following features.

1. Programmable functions

To retain the flexibility required for future expansion, the functions will be implemented in software rather than utilizing the rigid hard-wired ones.

2. Offline editing

This allows composing and editing data prior to sending it to storage. For example, an operator could

compose a message on the CRT screen using the keyboard, verify its accuracy, correct text if necessary, then go on the ring to transmit data to AUX.

3. Online editing

Smart Terminal will accept the abbreviated code for a particular line. It will initiate the message in the prescribed format and transfer to RI for transmission to AUX. It will accept a line or lines from AUX and display them on the terminal's screen. Data can now be reviewed, changed or updated if necessary, then returned to AUX for appropiate line-numbered storage. In addition to the function of line modification, entire lines can be inserted or deleted by a special hold and "Delete line" code within the AUX. This enables the operator to change, add, or delete a line and still maintain the required program continuity.

4. Message handling function

Smart Terminal will be capable of formulating messages to EP, thus requesting it to print a file or display messages from another processor or terminal. The same rapport will be available with AUX, enabling the ST to create, expand, or delete a file.

5. Miscellaneous functions

In addition to the above the ST will perform a continuous fetch, search, and return until the required item is discovered (e.g. request for the occurance of first "do" loop, or the name "Smith" in data).

III. REMOTE BATCH ENTRY SYSTEM PROTOCOL

An advantage of the RBES protocol is that it permits communications between support systems and the RBES without the sender actually knowing the physical location of the destination. The protocol will be standardized and independent of all process code structures. Messages will consist of the following fields:

1. Destination Code (8 bits)
2. Source Code (8 bits)
3. Text (command code/required data)
4. EOM code

This basic message format is delivered to the ring interface where the required prefix and suffix message structures are added to make it ring compatible. Since that procedure remains the same for all processors on the ring, it will be disregarded in the following descriptions.

A. SMART TERMINAL/AUXILIARY STORAGE UNIT PROTOCOL

Here the requirement narrows down to setting up a communications link between the Smart Terminal and an auxiliary storage unit for the purpose of creating, updating or deleting files.

1. Initial file access/creation

Initially, the ST will query the operator for his user number and file name. This will be used in the Access/Creation message. Figure 2. shows a message created in ST. Each AUX would receive the message and compare the user number/file name combination with those held in its "catalog." Whether or not it makes a match, it simply passes the message around the ring. However, if the combination is found the AUX sends an answer to the ST,

which will include the file's password in the text portion of the message.

The ST accepts the message (via input port of its MCS-8), stores the AUX address, and queries the user for the password. If a match is made the user can proceed with normal fetch/edit operations, if not, the ST concludes all interaction with the user. On the other hand, if there is no answer from any AUX (i.e. no such file in storage), then a request-for-storage message is automatically composed and sent to the AUX. When answered affirmatively, the ST query is made for a user-selected password, and composition of the file can begin.

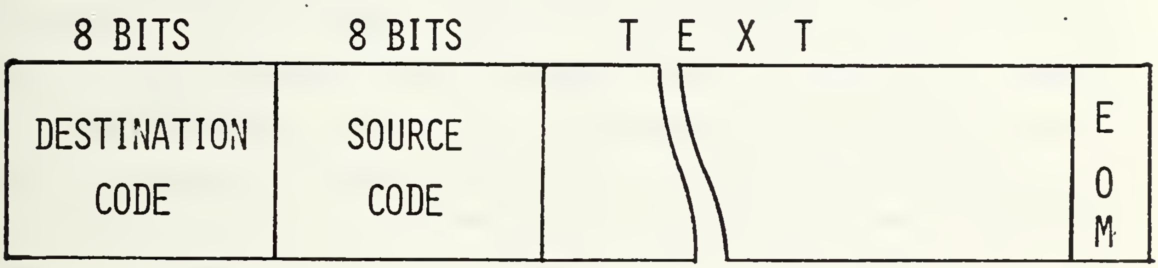


Figure 2. Message Format

2. File editing

When the ST detects a buffer full condition or an operator directive, a storage message is formed including the following fields:

1. AUX code (8 bits)
2. Sending unit code (8 bits)
3. Storage code
   a. Line Number/Data
   b. Line Number/Data
   c. etc.
4. ECM

The message is transmitted to the AUX and the data is stored in the appropriate file. When access to a particular line is desired, a similar message is composed, with a fetch code and line number placed in the text field. Data is fetched a line at a time. AUX retains a copy of the line requested and replaces it when a store message specifying the same line number is received. To delete a line, the ST transmits a line delete code and line number in a message to the AUX. In turn the AUX then tags the line with a special delete code but does not, at this time, erase it. Nevertheless, as far as the user or ST is concerned, the line ceases to exist once the acknowledge message is received. If at any time in the editing process the user wishes to insert a line, an insert message is sent with line number to be inserted. The AUX then holds all lines received thereafter in temporary storage until an end-of-insert message at which time they are reindexed. When the AUX inserts lines, those that have been coded "deleted" will be overwritten.

3. File Deletion

To preclude any malicious or inadvertent erasure of files, the deletion procedure will send a request message (delete code/file name) to the AUX. It in turn will acknowledge and require an additional message containing the correct password for completion. Any erroneous password will terminate the deletion procedure. At deletion, the AUX deletes the file name and makes that storage area available.

A confirming message is sent to the requesting ST which in turn informs the user the name of the file that was deleted.

4. File Expansion

The only communication between ST/AUX not initiated by the ST is that of file expansion. When the allotted file space reaches a predetermined point, the AUX will look into itself for more file space (i.e. another allotment), and if found it will annex it to the existing file space. If it does not have any available space it may send a message to another AUX requesting additional space and then link to it. If there is no space available the AUX will send a warning message to ST, informing it of impending space shortage. An entry in the catalog within each AUX might be used to limit the amount of space available to a user.

5. Sign-off

At sign-off the ST will request that the AUX "close" the file, preserving its contents for future access by ST or BP.

B. SMART TERMINAL/BATCH PROCESSOR PROTOCOL

Once the program has been formulated to user satisfaction and the ST/AUX link severed, a "run" message is sent to the BP requesting that a file be read and executed. The file will presumably contain any control cards needed by the BP (i.e. OS/360 JCL) along with optional program and data cards. The BP will acknowledge the message when file is located. Once received, the user can secure the ST or reinitialize for another program without fear of losing the file being executed. At job completion, BP will transfer output data to the AUX and signal ST that the job is ready. If the ST is secured, then information (Job complete/File name) goes into the Ring "Mail Box" process for later

retrieval by the ST. The protocol for "Mailbox" access will involve display, delete, and entry requests by the ST or BP and corresponding acknowledgement and "text" messages from the "mailbox" process.

C. BATCH PROCESS/AUXILIARY STORAGE UNIT PROTOCOL

Since the communication with the BP (i.e. access input files, create output files) appears to the AUX the same as that with the ST , then the protocol between the BP and AUX will be the same as that for ST/AUX.

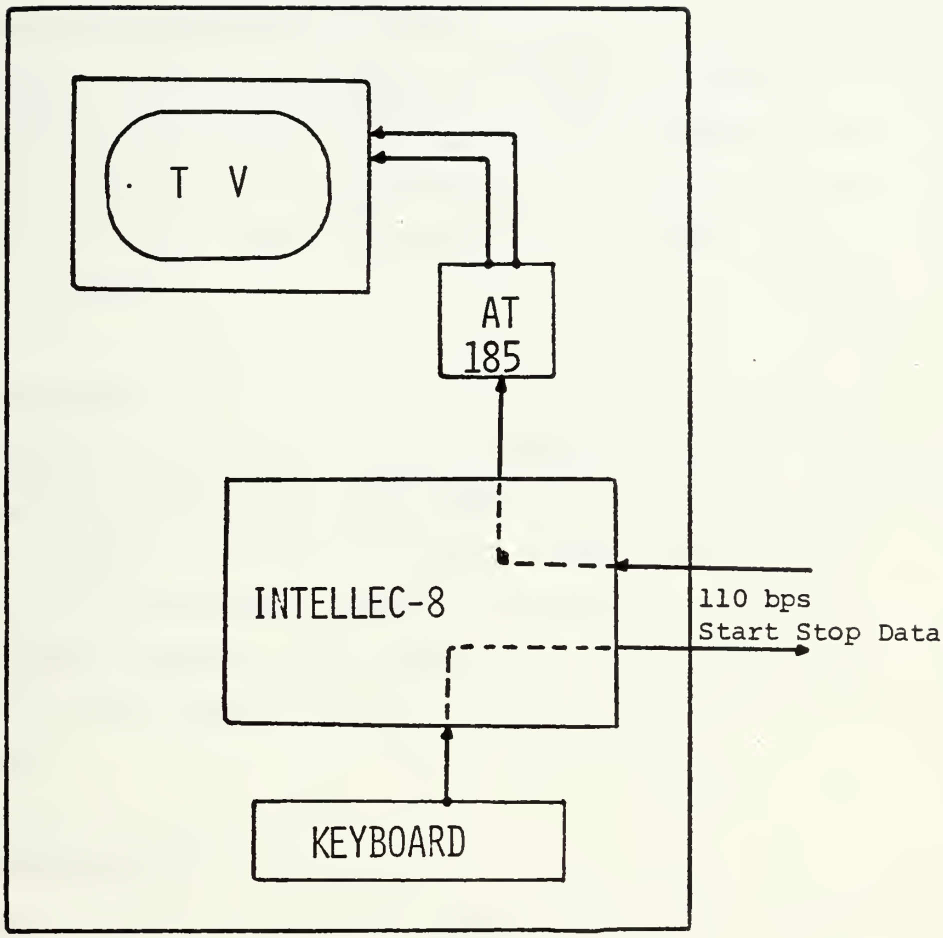


Figure 3. Prototype ST

IV. IMPLEMENTATION OF A PROTOTYPE SMART TERMINAL

A. COMPONENTS OF THE SMART TERMINAL

The Remote Batch Entry System is built around a series of Smart Terminals. In the prototype system each ST consists of a Keyboard, a Display Unit, and an Intellec-8 microcomputer system. The prototype of this system is depicted in Figure 3.

1. Keyboard

The RBES keyboard is diode decoded into the 8 bit (7 bit data and 1 parity bit) ASCII code. It has a standard typewriter layout with 26 upper case alphabetics, 10 numerals, and the standard special characters. Control and shift are each recognized by their own individual bit. The function of this keyboard is to provide operator entry of message data.

2. Display Unit

The display section of the RBES employs a standard, unmodified television equipped with the usual adjustments. The TV is connected to the interface via its antenna leads and operates on channel 2. The Intel Corporation allowed us to use the design for their prototype AT-185 TV interface in this project. Since the unit was designed to interface directly with the MCS-4 address/data bus, and the RBES will use the MCS-8, a few modifications were necessary. The interface requires +5 and -9 volt power supply. Six bits of data and eight bits of address are fed into the interface's memory with the READ/WRITE and DATA DISABLE signals generated by the MCS-8. This allows the MCS-8 to control

all the data displayed on the screen.

Characters are generated with the 5x7 dot matrix technique. The 64 character set consists of 26 upper case alphabetics, 10 numerals and the standard special characters. Each of the 256 words in memory is made up of 6 bits, one from each of the 6 memory chips. Each memory chip has 256 words, where each word is 1/6 of a character. This produces a display of 8 lines, 32 characters per line.

3. Intellec-8

The heart of the RBES is the Intellec-8 microcomputer system, consisting of a programmable read only memory (PROM) module, a CPU module, an input/output module and a random access semiconductor memory (RAM) module. All data from the keyboard and to the display is passed through the Intellec-8 for software processing.

B. EXPLANATION OF THE PROGRAM

Phase two of the project, to operate the ST as a full duplex TTY, has been completed. The program for that implementation is included in this thesis and appears in Appendix 1. It was written in PLM and compiled into MCS-8 object code on the IBM-360/67 via CP/CMS. The RBES can duplicate all functions of a full duplex TTY. Data in start-stop format is received via the UART (Universal Asynchronous Receiver Transmitter chip) and displayed on the TV screen just as it would be typed by a teletype printer. Data entered through the RBES keyboard is sent to the UART and transmitted in start-stop format.

V. CONCLUSIONS AND RECOMMENDATIONS

The proposed system is still in the initial experimental stages, and consequently does not contain all the refinements of a production system. The design parameters outlined in this report coupled with the success of the TTY-like prototype strongly attest to the feasibility and practicality of such a system. The program illustrated not only demonstrates the ST's basic abilities, but offers to any would-be successor a modularized development tool, through which each function of the "Smart Terminal" described may be individually developed, tested and incorporated. Given further development time the RBES concept should fulfill most expectations and still be economically feasible. The potential versatility of this system is its strongest justification.

APPENDIX 1

The main section of the TTY program is a continuous checking of the control lines, DATA AVAILABLE (a UART signal indicating an 8-bit character has been received and buffered) and TRANSMIT BUFFER EMPTY (a UART signal indicating buffer is available to accept and then transmit the next character). Positive response from either of these during the check cycle causes a series of subroutines to be called. The first results in the input data being displayed or screen being changed, while the other results in data being transmitted from the keyboard. The following is an explanation of the subroutines which can be called from each of the respective checks.

1. <u>DATA AVAILABLE</u>

A. GNC (Get Next Character) fetches character from the UART and determines if it is a character to be displayed or a special code to be carried out.

B. SPHA (Special Handling) is called by GNC and decides if the special code is a line feed or a carriage return and carries out the required command. If neither, it ignores the code.

C. DISPLAY keeps count of the position of the next character to be displayed on the bottom line. It takes that character and loads it into the Internal Buffer and the TV interface's memory at the address counted.

D. SCROLL is called by main when the line is full. It relocates all data on screen up 1 line and blanks out the bottom line.

2. <u>TRANSMIT BUFFER EMPTY</u>

A. KNC (Key Next Character) checkes to see if a key

has been depressed and if so it readjusts it into standard ASCII code for transmission.

B. SNC (Send Next Character) takes the data and sends it to the UART for transmission in start-stop format.

PLM TTY PROGRAM

```
 /*
          GLOBAL VARIABLES
*/
DECLARE (BCN,PTR,RDY,UNKN,ZIP)BYTE;
DECLARE BUF (256) BYTE;
DECLARE CR LITERALLY '0DH',
        LF LITERALLY '0AH';
/*
          CLEAR THE INTELLEC'S BUFFER
*/
CLEAR: PROCEDURE;
       DECLARE I BYTE;
       DO I = 0 TO 255;
          BUF(I) = 32;
       END;
       RETURN;
       END CLEAR;
/*
          DISPLAY TAKES THE CHARACTER AND ITS ADDRESS,
          STORES IT IN INTELLEC'S MEMORY, WRITES IT IN
          INTERFACE'S MEMORY FROM WHICH IT IS DISPLAYED.
          THE ADDRESS PLUS ONE IS RETURNED TO CALLER.
*/
DISPLAY: PROCEDURE(CHAR,ADR)BYTE;
       DECLARE(CHAR,ADR) BYTE;
       BUF(ADR) = CHAR;
       OUTPUT(2) = 191;    /* DISABLES SCREEN */
       OUTPUT(3) = NOT ADR;/* MEMORY LOCATION */
       OUTPUT(2) = CHAR AND 63;/* CHARACTER IS WRITTEN */
       OUTPUT(2) = 192;    /* SCREEN ON WITH NEW CHARACTER*/
       ADR = ADR + 1;      /* ADDRESS OF NEXT CHARACTER */
       RETURN ADR;
```

```
        END DISPLAY;
/*
            SCREEN DISPLAY IS MOVED UP ONE LINE, BOTTOM LINE
            WILL NOW BE BLANK.
*/
SCROLL: PROCEDURE;
        DECLARE I BYTE;
        I = 0;
        DO WHILE I < 224;
            I = DISPLAY(BUF(I+32),I);
            END;
        DO WHILE I > 0;
            I = DISPLAY(' ',I);
            END;
        ZIP = ZIP-32;
        RETURN;
        END SCROLL;
/*
            THIS PROCEDURE  PROCESSES ALL SPECIAL CHARACTERS
*/
SPHA:   PROCEDURE(SPC);
        DECLARE SPC BYTE;
        IF SPC = CR THEN
            DO; PTR = ZIP+1;
            RETURN;
            END;
        IF SPC = LF THEN
            DO; CALL SCROLL;
            ZIP = 224;
            PTP = DISPLAY('>',ZIP);
            RETURN;
            END;
        RETURN;
        END SPHA;
/*
            FETCH DATA FROM INPUT PORT 0 AND DETERMINE IF IT
```

```
         SHOULD BE DISPLAYED OR GET SPECIAL HANDLING.
*/
GNC:   PROCEDURE (X) BYTE;
       DECLARE X BYTE;
       X = (NOT INPUT(0) AND 07FH);
       IF X < 20H THEN
          DO; CALL SPHA(X);
          RETURN 0;
          END;
       UNKN = X;
       RETURN 1;
       END GNC;
/*
          KNC READS THE UNIQUE CODE OF THIS KEYBOARD THEN
          TRANSFORMS IT INTO STANDARD ASCII FOR TRANSMISSION
*/
KNC:   PROCEDURE (Y) BYTE;
       DECLARE Y BYTE;
       IF INPUT(2) > 0 THEN  /* INDICATOR OF KEY DEPRESSED*/
          DO; Y = INPUT(2);  /* DATA READ */
          UNKN  = (Y AND 07FH);  /* PARITY BIT REMOVED */
          IF (ROL(INPUT(1),1)) THEN  /* SHIFT KEY PUSHED */
             DO; IF UNKN = 40H THEN UNKN = 0;
             RETURN 1;
             END;
          ELSE DO; IF UNKN > 3FH THEN RETURN 1;
               IF UNKN > 3BH THEN
                 DO; DO CASE (UNKN-3CH);
                       UNKN = 2CH;
                       UNKN = 2DH;
                       UNKN = 2EH;
                       UNKN = 2FH;
                     END;
                 RETURN 1;
                 END;
               IF UNKN > 2BH THEN
```

```
                    DO; DO CASE (UNKN-2BH);
                         UNKN = 3CH;
                         UNKN = 3DH;
                         UNKN = 3EH;
                         UNKN = 3FH;
                        END;
                    RETURN 1;
                    END;
                  IF UNKN = 20H THEN RETURN 1;
                  IF UNKN > 20H THEN
                    DO UNKN = UNKN+16;
                    RETURN 1;
                    END;
                  IF UNKN = CR THEN RETURN 1;
                  IF UNKN = 1EH THEN
                    DO; UNKN = LF;
                    RETURN 1;
                    END;
                  END;
        RETURN 0;
        END KNC;
/*
           THE OUTPUT PORT IS LOADED WITH CHARACTER TO BE
           TRANSMITTED
*/
SNC:    PROCEDURE(WORD)BYTE;
        DECLARE WORD BYTE;
        OUTPUT(0) = NOT WORD;
        RETURN WORD;
        END SNC;
    PTR = 225;
    ZIP = 224;
    CALL CLEAR;  /* MEMORY IS FILLED WITH BLANKS */
    CALL SCROLL; /* SCREEN IS CLEARED */
/*
           CHECK IF DATA TRANSMISSION LINE CLEAR
```

```
*/
  MAIN: IF NOT(ROR(INPUT(1),2)) THEN
             DO; RDY = KNC(UNKN);
             IF RDY = 0 THEN BCN = 1;
             IF (RDY AND BCN) THEN
                DO; BCN = 0;
/*
           CHECK   FOR DEPRESSION OF  "CONTROL" KEY
*/
                IF (ROL(INPUT(1),2)) THEN UNKN=UNKN AND 01FH;
                UNKN = SNC(UNKN);
                END;
             END;
/*
           CHECK   IF DATA AVAILABLE LINE READY
 */
        IF NOT(INPUT(1)) THEN
           DO; RDY = GNC(UNKN);
           IF RDY THEN PTR = DISPLAY(UNKN,PTR);
           END;
        IF PTR = 0 THEN
           DO; CALL SCROLL;
           PTR = 224;
           END;
        GO TO MAIN;
EOF
```

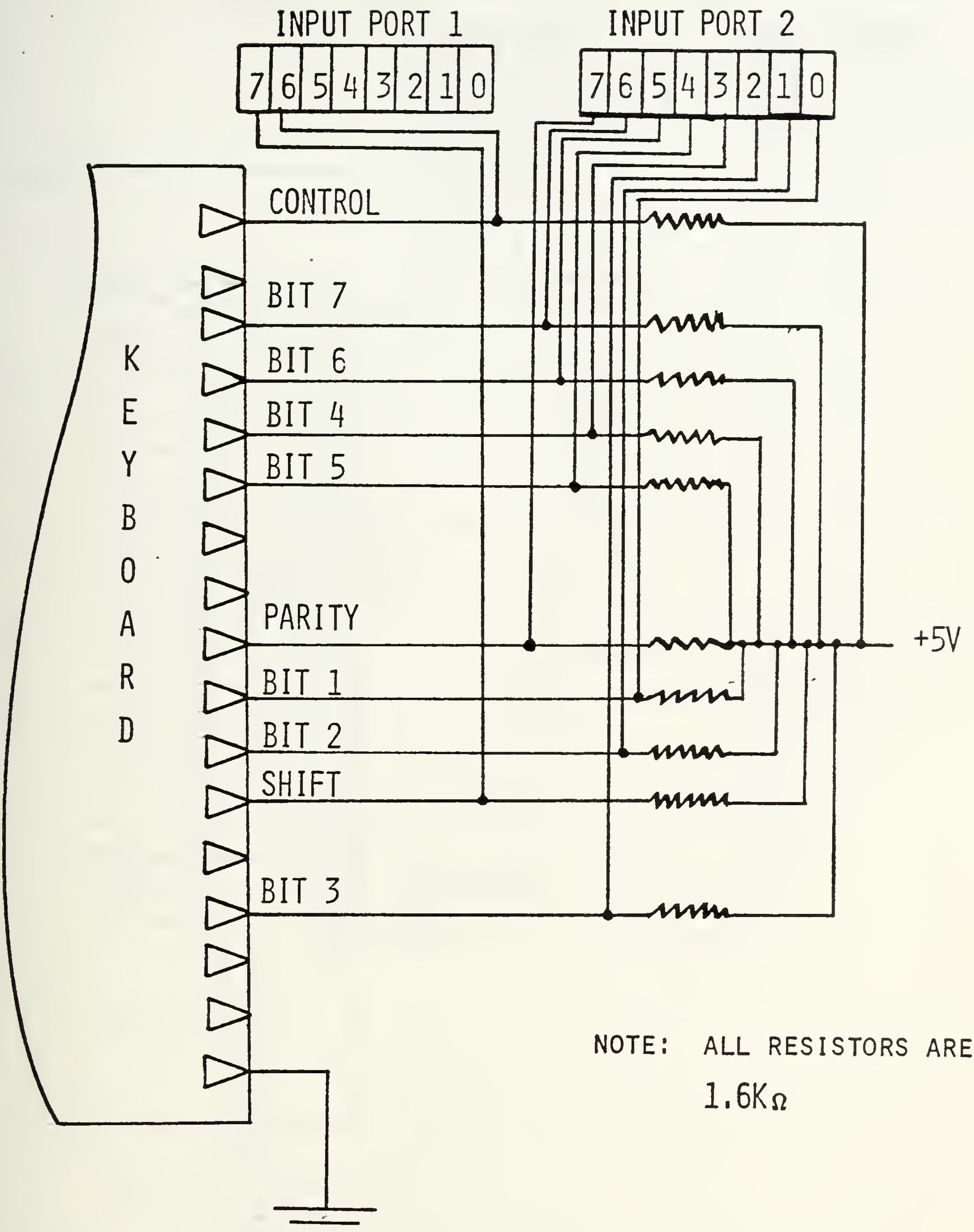
APPENDIX 2

The major modification to the AT-185 Interface involved the following:

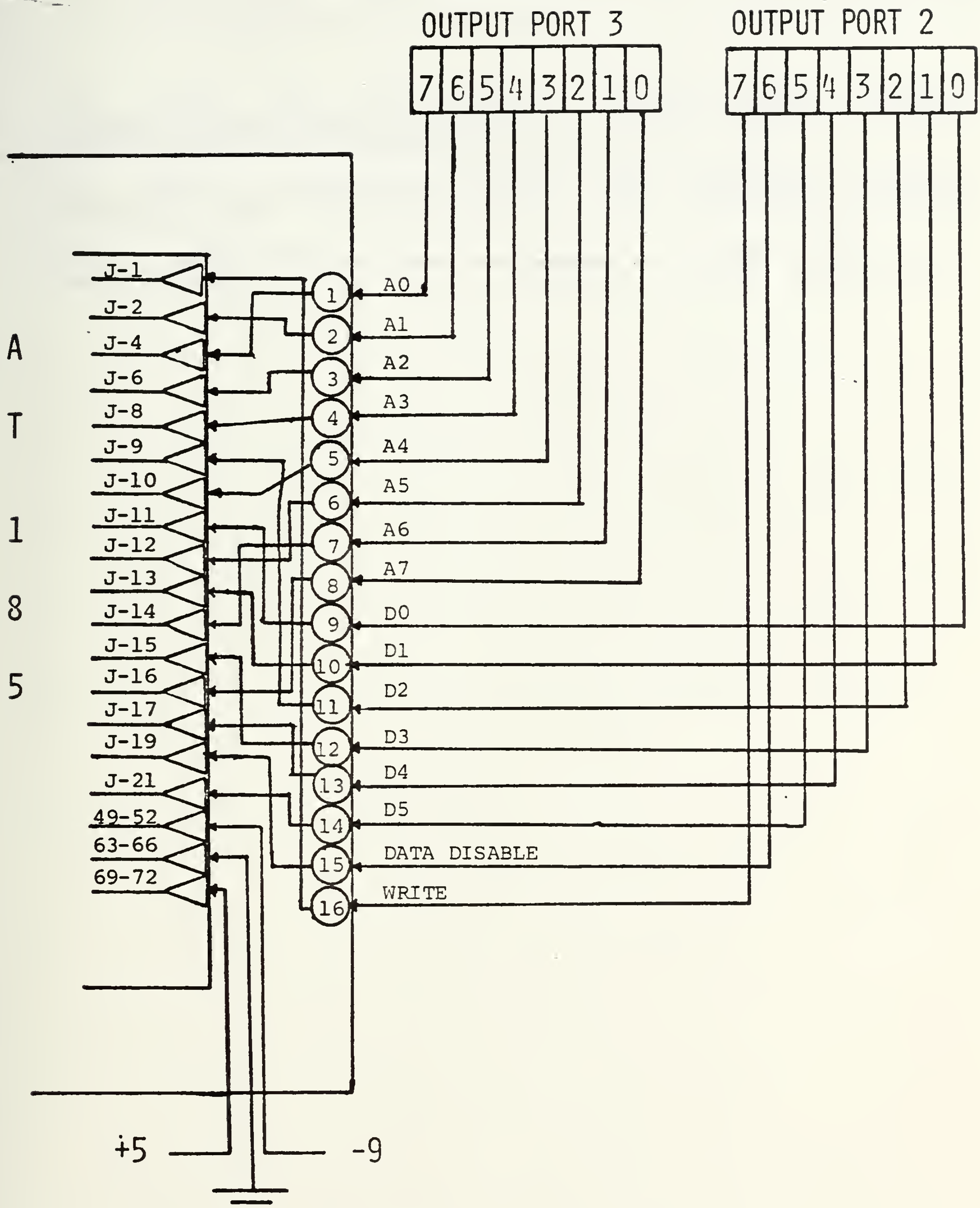
1. Remove the 4008 I.C. and its corresponding Resistors.

2. Connect Data and Address lines directly to edge Connector.

Now the address and data ports can be accessed directly by the INTELLEC-8. The following Figures show the required connections to various components of the prototype ST.



KEYBOARD TO INTELLEC-8



TV INTERFACE TO INTELLEC-8

BIBLIOGRAPHY

1. Hirt, K.A., _A Prototype Ring-Structured Computer Network Using Micro-Computers,_ Masters Thesis, Naval Postgraduate School, Monterey California, 1973.

2. INTEL Corporation, "8008 8-Bit Parallel Central Processor Unit Users Manual," _MCS-8 Micro-Computer Set,_ November 1973.

3. INTEL Corporation, _A Guide to PL/M Programming,_ September 1973.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
2.	Chairman Computer Science Group (Code 72) Naval Postgraduate School Monterey, California 93940	1
3.	Asst. Professor Ray Brubaker (Code 72Bh) Computer Science Group Naval Postgraduate School Monterey, California 93940	1
4.	ENS Gary Raetz (Code 72Rr) Computer Science Group Naval Postgraduate School Monterey, California 93940	1
5.	LCDR Richard J. Simpson USN Attack Squadron 42 N. A. S. Oceana Virginia Beach, Virginia 23240	1

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)

Remote Batch Entry System Utilizing Microcomputer and Television Display

5. TYPE OF REPORT & PERIOD COVERED

Master's Thesis; March 1974

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)

Richard John Simpson

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Naval Postgraduate School
Monterey, California 93940

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

Naval Postgraduate School
Monterey, California 93940

12. REPORT DATE

March 1974

13. NUMBER OF PAGES

35

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

Naval Postgraduate School
Monterey, California 93940

15. SECURITY CLASS. (of this report)

Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

microcomputer
terminal
ring network
T. V. display

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A project to develop a ring telecommunication network is currently underway at the Naval Postgraduate School. A remote batch entry system is being developed as a part of that project. This paper investigates the protocol for such a system and proposes a smart CRT terminal for use as a remote station. It first develops the requirements for such a terminal and then presents the actual implementation of a TTY-like CRT terminal as the foundation in the development of a smart terminal.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
(Page 1) S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DD Form 1473 (BACK)
1 Jan 73
S/N 0102-014-6601

150327

Thesis
S5125 Simpson
c.1 Remote batch entry system utilizing micro-computer and television display.